Stability Augmentation and Fault Tolerance for a Hexapod Underwater Vehicle

Olivia Chiu[*1], Meyer Nahon[*1], Nicolas Plamondon[3]

Centre for Intelligent Machines, McGill University
3480 University St., Montreal, Canada, H3A 2A7.

[1]omychiu@gmail.com; [*2]Meyer.Nahon@mcgill.ca; [3]Nicolas.Plamondon@pwc.ca

Abstract

AQUA is an underwater hexapod robot that uses paddles to propel and orient itself. The system is typically operated remotely by a pilot, with feedback from cameras and on-board sensors. In this work, a stability augmentation system was developed and evaluated on the robot. In order to study the stability of the system, its model was linearized about a nominal equilibrium at several different cruising speeds. Since the robot is never truly in equilibrium due to its oscillating paddles, this required a novel approach. The stability of the unaugmented vehicle was evaluated and improved using sensor feedback. The stability augmentation system was then modified to compensate for possible faults that could occur during the operation of the robot. The failure of a leg was investigated by analyzing the additional drag forces created by the fault. The controller was implemented on the robot with encouraging results.

Keywords

Underwater Vehicles; Biomimetic; Fault Tolerant Control; Stability Augmentation.

Introduction

AQUA is a six-legged amphibious robot, shown in FIG. 1, which can swim with the use of oscillating paddles. With these flippers, the robot is able to directly control roll, pitch, yaw, surge and heave. This makes AQUA unique compared to other underwater vehicles which use thrusters to propel themselves.

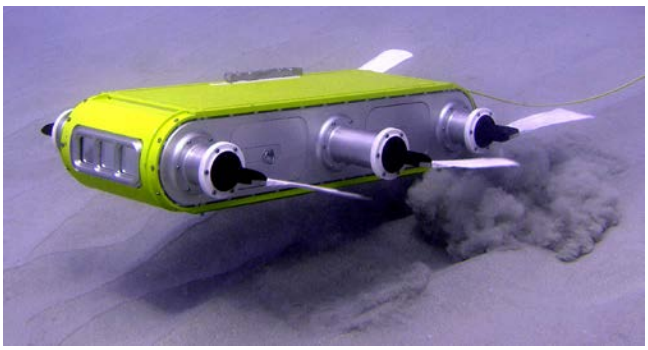

FIG. 1 THE AQUA HEXAPOD ROBOT

The environment in which AQUA operates is often unpredictable and can perturb the robot. One way to reduce the influence of the external forces on the system and thus stabilizing it, is through the use of a stability augmentation system (SAS). This type of system is widely used in flight control to aid pilots and to improve the response of a vehicle (Fullmer et al., 1992; Oliva, 1994; Kahn, 2003). Such systems are not as commonly found in underwater vehicles and the issues of stability are often incorporated into the design of the tracker rather creating a separate controller (Nakamura and Savant, 1992; Do et al., 2004; Licht et al., 2007).

Previous work on trajectory tracking controllers for 'conventional' underwater vehicles is extensive. A survey was done by Yuh (2000) on the design and control of Autonomous Underwater Vehicles (AUVs). These included sliding, nonlinear, adaptive, neural network and fuzzy control. Other experimental comparative studies have also been done by Lea et al. (1999) and Smallwood et al. (2004). They implemented different types of controllers on an AUV or ROV (Remotely Operated Vehicle) and compared the performance and complexity of each controller. This is by no means a complete list of all controllers available for underwater vehicles, however it must be emphasized that all the above controllers were designed for systems that utilized thrusters for propulsion. There have been very few studies into controlling foil-based vehicles, which include work by Hsu et al. (2003) and Licht et al. (2007).

Prior work on fault tolerance falls into two main categories: fault detection or control reconfiguration. Research on fault detection usually consists of comparing the behavior of the vehicle to a model and identifying discrepancies. A fault is detected when the discrepancies exceed a predetermined threshold (Orrick et al., 1994; Rae and Dunn, 1994). A common approach to compensating for faults is to design robust controllers (Fossen and Balchen, 1988; Leonard,

1995). However, if the controller is not robust enough, the control system may need to be reconfigured to compensate for the detected fault (Yang et al., 1999; Ni and Fuller, 2003).

The work presented here involves the design and evaluation of a stability augmentation system for the AQUA robot. It begins with a description of the vehicle and its operation. Next, a novel method of linearization of the system is used to study the system stability. The linearized model is then used to design and implement a stability augmentation system. Finally, faults that may occur are investigated and the adaptation of the stability augmentation system for fault compensation is discussed. The key contribution of this work is the investigation of these topics for a vehicle that uses oscillatory paddling propulsion.

Description of Vehicle and its Operation

The design of the AQUA underwater vehicle is based on the Rhex land-based robot (Saranli et al., 2001), which is a terrestrial six-legged robot. RHex's semicircular legs were replaced with flippers and the outer shell was redesigned such that it could survive in a water environment, creating AQUA (Georgiades et al., 2004). With these flippers, AQUA is able to directly control roll, pitch, yaw, surge and heave. The thrust created by the flippers can be regulated by changing the period or amplitude at which they oscillate and the direction of the thrust can be adjusted by changing the center position or offset of the oscillations. At the moment AQUA has a tether which allows for communication and data to be exchanged between the robot and a pilot.

The robot's motion is measured with a Microstrain 3-axis inertial measurement unit (IMU) that provides translational acceleration, angular velocity, roll angle and pitch angle. The yaw of the robot is measured by an on-board magnetometer-based compass from True North Technologies, whose accuracy is limited due to the surrounding electronics within the robot. The robot also has three Point Grey cameras on-board: two facing forward and one facing rearward.

Experiments are usually done in a McGill swimming pool or in the open sea at the McGill Bellairs Institute in Barbados. During experiments, AQUA is accompanied by two or three divers or swimmers who monitor the robot's safe operation. The McGill pool has 8 lanes and is 25 meters long. Most experiments take up the entire length of the pool but there is a preference to use the deeper end to accommodate diving maneuvers. At the Bellairs Institute, some experiments take place off the shore in shallow water and others are done in deeper waters. Experiments done by the shore are simpler to set up but AQUA then needs to contend with the surf when swimming and the visibility is often hindered by sand kicked up by the swimmers. In deeper waters, the water tends to be calmer under the surface and the visibility is better. However, the pilot must then sit in a boat to control the robot and the entire setup must be powered by Li-ion batteries, thus limiting the number of experiments and their durations.

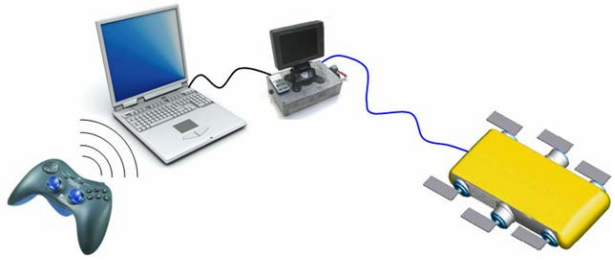

FIG. 2 THE GENERAL EXPERIMENTAL SETUP

FIG. 2 shows the general setup used when performing experiments. A 200m fibre optic tether is attached to the back end of the robot to allow for communication and data transfer between the robot and a pilot on land. The other end of the fibre optic is connected to an Operator Control Unit (OCU) which converts the signals between the fibre optic cable on the robot and the serial cable from the pilot's computer. On top of the OCU is a video screen that displays the images captured by any one of the three on-board cameras. Using this, the pilot is able to see where AQUA is swimming. It also allows any divers swimming with AQUA to signal the pilot if a problem occurs. The OCU is connected to the pilot's computer, which is used to monitor various sensors in AQUA, to send commands to the robot and to log the data associated with an experiment.

The pilot is able to do all this through a Graphical User Interface (GUI). The GUI allows the pilot to switch into the swimming mode of the robot and to calibrate the flippers. The GUI also displays the orientation of the robot and the angular commands given by the pilot and the controllers, as well as the oscillation period, amplitude and offset of the six paddles. The interface allows changing controllers and controller gains on the fly during an experiment. Data from the various sensors can be selected in a separate panel and logged as the robot performs an experiment. The GUI also includes a health monitor that shows the pilot the power consumption, leg positions and battery state.

While the pilot could control AQUA by moving the sliders in the GUI, it is much easier to drive the robot with a wireless gamepad, which is shown on the left side of FIG. 2. The gamepad sends the commands from the pilot to the computer; then to the OCU and finally to the robot. The left joystick controls the yaw and heave of the robot, while the right joystick controls the roll and pitch of the vehicle. The speed of the robot is dictated by the period and amplitude of the oscillations of the flippers. While it is possible to change these parameters during the operation of the robot, they are usually left unchanged throughout an experiment. The various buttons on the gamepad allow the pilot to select different modes (such as calibrate, stand or underwater) or to start or stop the robot's movements.

One issue that has been noted by AQUA pilots is the lack of stability while the robot swims. This was the motivation for the design of a stability augmentation system (SAS) to help stabilize the robot. This would facilitate the use of AQUA for surveillance and research of coral reefs or for hull inspections. The tasks undertaken by AQUA can be fairly lengthy and take place in hazardous conditions which could lead to failures of one or more of the flippers. Previous experiments with AQUA have experienced the loss of a flipper as well as a motor failure which caused a flipper to be stuck at some angle. This then served as motivation for our investigation of the fault tolerance under these conditions.

Dynamics Model

A dynamics model of the vehicle can be used to support the design of a stability augmentation system. The nonlinear model used, described in detail by Georgiades et al. (2009), is summarized and then linearized in the following sections.

Nonlinear Model

The robot has six degrees of freedom, and we consider two relevant reference frames. The first is a robot-fixed frame that has its origin at the center of mass of the robot. The second is an inertial reference frame that has its origin at a fixed arbitrary point on the water surface. Euler angles ($\phi\ \theta\ \psi$) are used to relate the orientations of the two coordinate frames, where ϕ is the roll angle, θ the pitch angle and ψ the yaw angle (Fossen, 1994). These are shown in FIG. 3 along with the numbering of the paddles. The motion of the robot can be described by $\mathbf{n}_1$, the translational position of the robot's mass center, $\mathbf{n}_2$, the Euler angles, and $\mathbf{v}$, the robot's generalized velocity:

$$\mathbf{n}_1 = [x\ y\ z]^T, \quad \mathbf{n}_2 = [\phi\ \theta\ \psi]^T, \quad \mathbf{v} = [u\ \ v\ \ w\ \ p\ \ q\ \ r\ ]^T \quad (1)$$

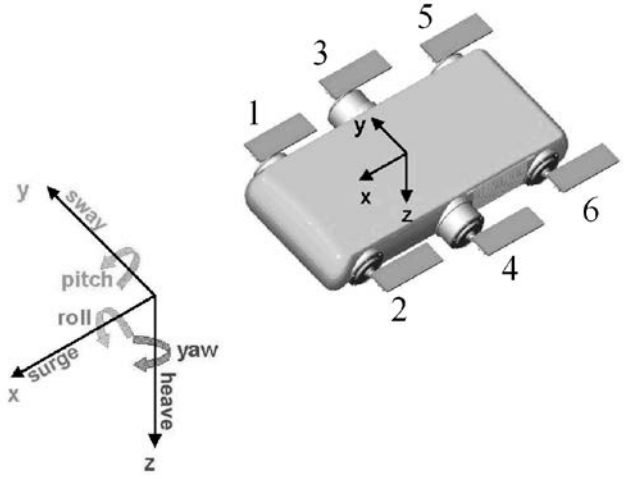

FIG. 3 DEGREES OF FREEDOM AND PADDLE NUMBERING

The position defined by $\mathbf{n}_1$ and $\mathbf{n}_2$ is expressed in the inertial reference frame while the velocity of the vehicle, given by components u, v, w, p, q, and r, is expressed in the robot frame. Transformation matrices are used to relate vectors in the two frames. The dynamics model of AQUA is developed based on a component breakdown approach and can be expressed as (Georgiades et al., 2009):

$$\mathbf{M}\,\dot{\mathbf{v}} + \mathbf{C(v)}\,\mathbf{v} + \mathbf{D(v)}\,\mathbf{v} + \mathbf{g}\,(\mathbf{n}_2) + \mathbf{b}\,(\mathbf{n}_2) = \mathbf{f} \quad (2)$$

where $\mathbf{f} = \begin{bmatrix} F_x & F_y & F_z & M_x & M_y & M_z \end{bmatrix}^T$ is the vector of net forces and moments produced by the paddles in the six degrees of freedom. The model developed by Georgiades et al. (2009) includes a comprehensive model of the paddle force generation, validated through experiments. **M** is the 6 x 6 mass matrix including added mass, **C(v)** is the 6 x 6 Coriolis matrix, **D(v)** is the 6 x 6 hydrodynamic matrix, **g** is the gravitational force vector, and **b** is the buoyancy force vector. In the simulation, it is assumed that the center of gravity is coincident with the center of buoyancy. As a result, the buoyant and gravity force cancel each other. In practice, they are never exactly coincident because the mass distribution changes depending on which batteries, set of paddles and other pieces of equipment are installed. Since the robot is immersed in water, the Coriolis and mass matrices include a rigid body and an added mass component. The rigid body part can be understood as the mass of the robot in a vacuum, while the added mass part models added

inertia due to the robot moving through the fluid. According to Fossen (1994), assuming that there are three planes of symmetry and that the vehicle is moving at low speed, the mass matrix including the rigid-body and added mass is diagonal. The hydrodynamic matrix is also a diagonal matrix whose elements represent the hydrodynamic forces and moments due to the robot's motion through the water. However, the Coriolis matrix **C(v)** has off-diagonal terms and is responsible for the coupling between the 6 degrees of freedom. Moreover, the Coriolis and hydrodynamic matrices contain the velocity vector. As a result, these two terms are responsible for the nonlinearity of the system. The parameters in the different matrices were obtained using empirical results for a solid rectangular prism (Fossen, 1994). With the paddle and robot motion known, (2) allows determination of the acceleration of the robot using

$$\dot{\mathbf{v}} = \mathbf{M}^{-1}[\mathbf{f} - \mathbf{C(v)}\,\mathbf{v} - \mathbf{D(v)}\,\mathbf{v} - \mathbf{g}\,(\mathbf{n}_2) - \mathbf{b}\,(\mathbf{n}_2)] \qquad (3)$$

The nonlinear dynamics model is useful for evaluating time histories of the robot's motion in response to particular paddle motion. When combined with the relevant kinematics relations, the non-linear model given in (3) can be written as (Georgiades et al., 2009):

$$\dot{\mathbf{x}} = \mathbf{g}(\mathbf{x}, \boldsymbol{\tau}) \qquad (4)$$

where $\mathbf{x}$ is the state vector $\mathbf{x} = [u \quad v \quad w \quad p \quad q \quad r \quad x \quad y \quad z \quad \phi \quad \theta \quad \psi]^T$. The vector $\boldsymbol{\tau}$ represents the propulsive force due to the paddles whose magnitude and direction depends on the period, amplitude and offset of oscillation of each paddle. It can be decomposed into x and z-components as $\boldsymbol{\tau} = [\tau_{x1} \quad \cdots \quad \tau_{x6} \quad \tau_{z1} \quad \cdots \quad \tau_{z6}]^T$, where τ_{xi} represents the force provided by paddle i in the body-frame x-direction and τ_{zi} in the z-direction.

Linearization

A disadvantage of the non-linear model is that it fails to provide quantitative information about the stability of the robot, and prevents the use of linear controller design theory. In order to overcome these drawbacks, the system was linearized to allow the eigenvalues and eigenvectors of the system to be studied. A linear model typically takes the form:

$$\dot{\mathbf{x}} = \mathbf{A}\mathbf{x} + \mathbf{B}\boldsymbol{\tau} \qquad (5)$$

where **A** is defined as:

$$\mathbf{A} = \frac{\partial \mathbf{g}}{\partial \mathbf{x}} = \begin{bmatrix} \frac{\partial g_1}{\partial x_1} & \frac{\partial g_1}{\partial x_2} & \cdots & \frac{\partial g_1}{\partial x_{12}} \\ \frac{\partial g_2}{\partial x_1} & \frac{\partial g_2}{\partial x_2} & \cdots & \frac{\partial g_2}{\partial x_{12}} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial g_{12}}{\partial x_1} & \frac{\partial g_{12}}{\partial x_2} & \cdots & \frac{\partial g_{12}}{\partial x_{12}} \end{bmatrix} \qquad (6)$$

and **B** is similarly defined as $\mathbf{B} = \partial \mathbf{g} / \partial \boldsymbol{\tau}$.

Several methods can be used to linearize a system. Numerical differentiation is a direct and common approach. It requires that the system first be in an equilibrium condition. A perturbation is then applied to element j of the state vector **x**, and the consequent response is evaluated. The elements of the state matrix **A** are then evaluated as $A_{ij} = \Delta g_i / \Delta x_j$, where g_i is the ith element of **g** and x_j is the jth element of **x.**

However, due to the oscillating paddles, the vehicle's velocity fluctuates periodically, even during steady-state level swimming. This implied that the robot was never in a clear equilibrium, and conventional linearization techniques could not be used directly. An additional issue is that the response to a disturbance depended on the position of the paddle. Thus, to deal with the 'oscillating equilibrium', an impulse disturbance was applied at various instants over one period of oscillation to get every possible paddle configuration. This can be seen in FIG. 4 for the case of a disturbance of 0.2 m/s in the first state variable u for the nominal steady-state forward velocity condition of 0.16 m/s.

With this approach, the response of the system for a wide variety of paddle configurations was obtained. These responses were then averaged over all the configurations and each entry of matrix **A** was calculated using $A_{ij} = \overline{\Delta g_i} / \Delta x_j$, where Δx_j is the disturbance, and the overbar denotes an average value of Δg_i obtained by subtracting the response with disturbance from the response without disturbance. By taking the average, the entries of **A** represented the average dynamics of the robot over one period of oscillation. Since the dynamics of the robot change drastically with velocity, evaluating **A** at a single operating point would not fully model the robot accurately. Thus, this procedure was applied at three different velocities (0.16m/s, 0.49m/s and 0.75m/s) that spanned the operating range of the robot.

The **B** matrix in (5) was obtained using a similar approach: the components of $\boldsymbol{\tau}$ were perturbed at various instants over one period of oscillation and Δg_i

was evaluated. In contrast to the elements of **A**, it was found that the entries of **B** were not time-periodic and also remained the same for all surge velocities.

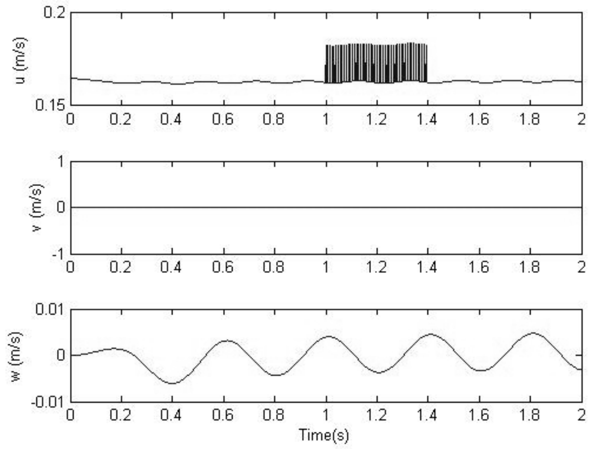

FIG. 4 TRANSLATIONAL VELOCITY OF CENTER OF MASS WITH DISTURBANCE IN SURGE VELOCITY

The validity of the state matrix **A** was verified by comparing the response of the linear and nonlinear models to perturbations in the initial conditions. This was done by giving an initial perturbation of 0.1m/s to the velocity variables from the initial condition. An example of this comparison is shown in FIG. 5.

As can be seen in the figure, there is a good agreement between the linear and nonlinear model---the linear model yielded an average response consistent with the nonlinear model, without the fluctuations due to the paddle oscillations. However, the linear model did not match the nonlinear model very well for disturbances in the lateral degrees of freedom (sway, roll and yaw in FIG. 3). The linear system was unstable in the lateral degrees of freedom while the nonlinear model was not.

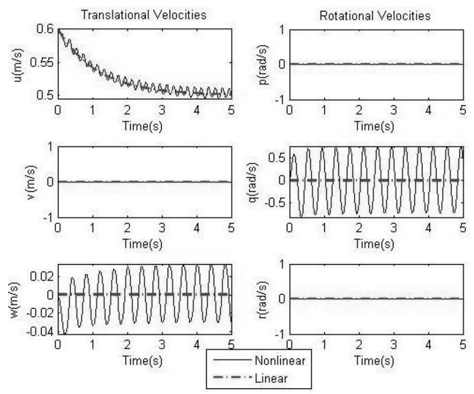

FIG. 5 COMPARISON OF LINEAR AND NONLINEAR SIMULATION RESULTS FOR STEADY-STATE VELOCITY OF V=0.5m/s AND INITIAL DISTURBANCE OF Δu=0.1m/s

The system described by (2) is coupled and highly nonlinear, but the linearization does not incorporate the coupling between the degrees of freedom which apparently stabilizes the nonlinear model.

SAS Design

Once the state space representations at three different forward velocities had been obtained, a stability augmentation system (SAS) could be designed. A stability augmentation system differs from an autopilot in that it does not ensure that the robot follows a trajectory. Rather, it aims to return all state perturbations to zero, thus reducing the impact of external disturbances on the system. This can be done by closing the feedback loop in the system and returning the measured states of the robot to the controller as seen in the Augmented System block of FIG. 6.

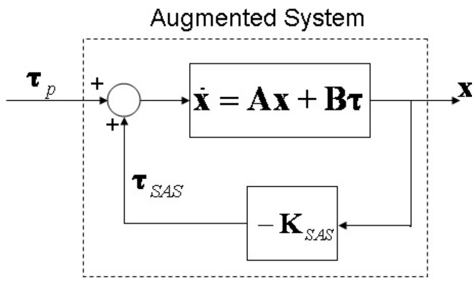

FIG. 6 BLOCK DIAGRAM OF THE STABILITY AUGMENTATION SYSTEM

In this figure, the control input $\boldsymbol{\tau}_{SAS}$ is provided by the SAS, while the input $\boldsymbol{\tau}_P$ is provided by the pilot or autopilot. The general notion is that the high-level controller 'sees' a new augmented system that is more stable than the original unaugmented system (with $\boldsymbol{\tau}_{SAS}$ = **0**). The SAS can be a proportional one and take the form

$$\boldsymbol{\tau}_{SAS} = -\mathbf{K}_{SAS}\mathbf{x} \quad (7)$$

where $\mathbf{K}_{SAS}$ is a 12x12 gain matrix of the form:

$$\begin{bmatrix} 0 & 0 & 0 & 0 & 0 & -K_r & 0 & 0 & 0 & 0 & 0 & -K_\psi \\ 0 & 0 & 0 & 0 & 0 & K_r & 0 & 0 & 0 & 0 & 0 & K_\psi \\ 0 & 0 & 0 & 0 & 0 & -K_r & 0 & 0 & 0 & 0 & 0 & -K_\psi \\ 0 & 0 & 0 & 0 & 0 & K_r & 0 & 0 & 0 & 0 & 0 & K_\psi \\ 0 & 0 & 0 & 0 & 0 & -K_r & 0 & 0 & 0 & 0 & 0 & -K_\psi \\ 0 & 0 & 0 & 0 & 0 & K_r & 0 & 0 & 0 & 0 & 0 & K_\psi \\ 0 & 0 & 0 & -K_p & -K_q & 0 & 0 & 0 & 0 & -K_\varphi & -K_\theta & 0 \\ 0 & 0 & 0 & K_p & -K_q & 0 & 0 & 0 & 0 & K_\varphi & -K_\theta & 0 \\ 0 & 0 & 0 & -K_p & 0 & 0 & 0 & 0 & 0 & -K_\varphi & 0 & 0 \\ 0 & 0 & 0 & K_p & 0 & 0 & 0 & 0 & 0 & K_\varphi & 0 & 0 \\ 0 & 0 & 0 & -K_p & K_q & 0 & 0 & 0 & 0 & -K_\varphi & K_\theta & 0 \\ 0 & 0 & 0 & K_p & K_q & 0 & 0 & 0 & 0 & K_\varphi & K_\theta & 0 \end{bmatrix} \quad (8)$$

where K_p, K_q, K_r, K_ϕ, K_θ, and K_ψ are the gains acting on the roll rate, pitch rate, yaw rate, roll angle, pitch

angle and yaw angle, respectively. The columns of the gain matrix $\mathbf{K}_{SAS}$ are associated with the elements of the state vector $\mathbf{x}$. The rows are associated with the propulsive forces given by the elements of $\boldsymbol{\tau}$ where the first 6 rows (one for each paddle) are related to forces in the x-direction, while the last 6 rows correspond to forces in the z-direction. To ensure that the effort is evenly distributed amongst the paddles, the entries in $\mathbf{K}_{SAS}$ associated with a particular state and a particular direction have equal magnitudes. Negative signs were assigned to some elements of $\mathbf{K}_{SAS}$ to account for the fact that the paddle force on one side of the robot needs to be opposite the force on the other side in order to counter certain disturbances. For example, to correct a yaw disturbance, the three flippers on one side of the robot need to create an x-force opposite to the flippers on the other side.

The control input $\boldsymbol{\tau}$ is made up of a pilot contribution and a contribution from the SAS, i.e. $\boldsymbol{\tau} = \boldsymbol{\tau}_P + \boldsymbol{\tau}_{SAS}$. If (7) is substituted into (5), we obtain

$$\dot{\mathbf{x}} = \left(\mathbf{A} - \mathbf{B}\mathbf{K}_{SAS}\right)\mathbf{x} + \mathbf{B}\boldsymbol{\tau}_p \qquad (9)$$

This new or augmented system, represented by $\mathbf{A} - \mathbf{B}\mathbf{K}_{SAS}$, is more stable, allowing the pilot or autopilot to focus on following a trajectory without being concerned with stabilizing the vehicle.

Eigenvalues

The stability of a linear system can be evaluated by examining at the eigenvalues of the $\mathbf{A}$ or $\mathbf{A} - \mathbf{B}\mathbf{K}_{SAS}$ matrix --- i.e. by solving

$$\det\left|s\mathbf{I} - \left(\mathbf{A} - \mathbf{B}\mathbf{K}_{SAS}\right)\right| = 0 \qquad (10)$$

Since the eigenvalues are also the poles of the system, any eigenvalue with a positive real part would indicate instability in the system. By analyzing the eigenvectors that correspond to the positive eigenvalues, it is possible to see which states are likely to become unstable.

This was done with the three $\mathbf{A}$ matrices derived in the previous section (one for each of three forward speeds) and setting $\mathbf{K}_{SAS} = \mathbf{0}$. At each velocity, there was only one eigenvalue with a positive real part, six zero eigenvalues and the remaining five had negative real parts. The eigenvector corresponding to the positive real eigenvalue was non-dimensionalized and normalized. The elements with larger magnitudes are more significant and are the states that will go unstable. It was found that regardless of the velocity of the robot, ψ and r were the most unstable states, followed by ϕ, p, θ and finally q.

Gain Selection

The gains of matrix $\mathbf{K}_{SAS}$ were determined in the same order when designing the SAS. The range of gains that stabilized each of the states was found by solving for $\mathbf{K}_{SAS}$ in (10) such that the eigenvalues of $\mathbf{A} - \mathbf{B}\mathbf{K}_{SAS}$ all had negative real parts. Within this range, trial and error was then used to determine what gains would reduce each state perturbation to zero.

Since ψ was the first state to go unstable, it was the first state to be fed back. This was done by setting all the elements in $\mathbf{K}_{SAS}$ to zero except for K_ψ. Solving (10) such that all poles have negative real parts, it was found that for V = 0.16m/s, $K_\psi > 0$ could stabilize the unstable mode. K_ψ was then assigned the value 1 N/rad, which reduced the yaw angle to zero in steady state when implemented in the linear model. However, with this feedback in the linear model, the roll and pitch angles were not close to zero at steady state and the yaw rate had some oscillations in it. Thus, keeping K_ψ = 1N/rad in the $\mathbf{K}_{SAS}$ matrix, the gain for the yaw rate was found using the same method by solving (10). Similarly, the roll and pitch gains were found and included one at a time in the $\mathbf{K}_{SAS}$ matrix using the same method as with the yaw angle. A similar process was used to create the gain matrices for V = 0.49m/s and V = 0.75m/s. With the SAS implemented in the linear system, the poles of the closed-loop system were now found to all be stable.

Performance

The gain matrices were then implemented in a closed-loop linear model of AQUA's behavior using MATLAB, and the results showed that the effects of disturbances were greatly reduced. To further evaluate the system performance, the same SAS was then implemented in the non-linear model which showed poorer responses than those seen in the linear simulations. The system became a little more stable at the lowest velocity but at higher velocities the system became unstable with the SAS.

Further refinement of the gain matrices for each of the steady state velocities was done by adjusting the gains for each of the states separately. The states were stabilized in the same order as with the linear system, beginning with yaw angle. The largest changes were made to the gains associated with ψ, θ and r. Some of the gains were up to 6 times higher than those used in the linear system and at higher velocities non-zero

gains were needed to control the pitch rate in order to keep the system stable.

FIG. 7 displays the angular and forward velocity responses from the nonlinear model at V = 0.49m/s. In this simulation 0.7 Nm moment disturbances (in the roll(a), pitch(b) or yaw(c) direction) were applied at t = 10s, lasting for 5 seconds. FIG. 8 shows the corresponding forward velocity response. The simulation indicates that the gain matrix used was effective in moderating the effect of the disturbances. The angles and forward velocity returned to the same values they had before the disturbance was applied.

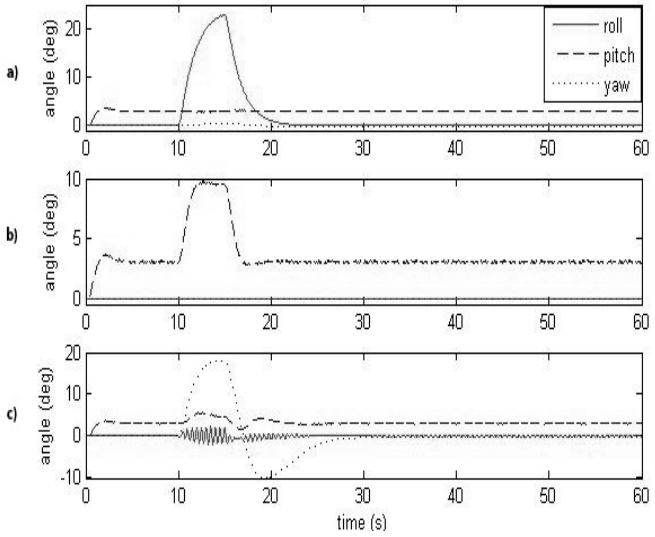

FIG. 7 RESPONSE TO ANGULAR DISTURVANCES APPLIED IN (a) ROLL, (b), PITCH, (c) YAW DIRECTIONS AT STEADY-STATE VELOCITY V=0.49m/s

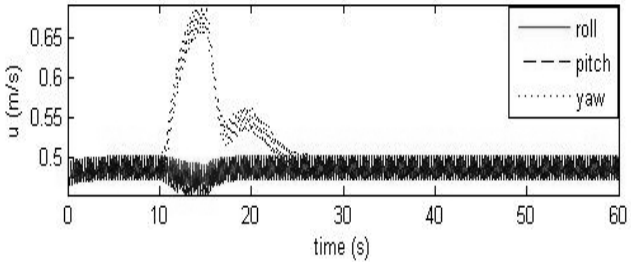

FIG. 8 FORWARD VELOCITY RESPONSE TO ANGULAR DISTURVANCES AT STEADY-STATE VELOCITY V=0.49m/s

Gain Scheduling

With the SAS designs discussed in the preceding sections, it was now possible to keep the system stable for three distinct forward velocities. However, during normal operation, the robot would function at all speeds within this range. Therefore, a gain schedule was needed to determine a suitable gain matrix to use at a given velocity. A simple gain schedule was implemented that interpolated linearly between the three gain matrices designed above. This gain scheduled SAS was then evaluated, as shown in FIG. 9.

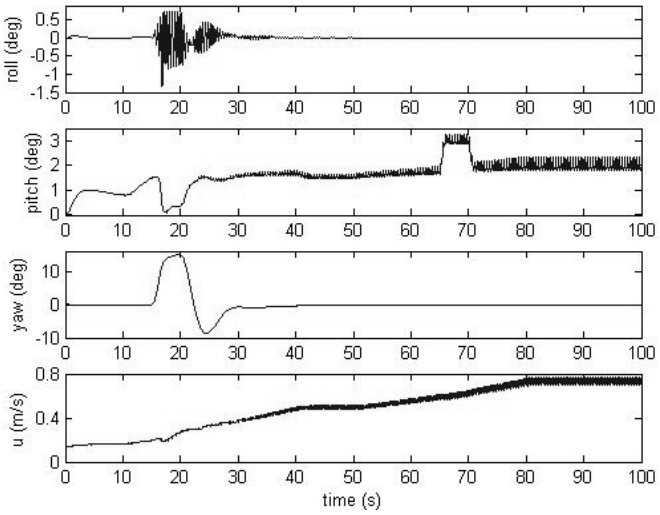

FIG. 9 ANGULAR RESPONSE AND FORWARD VELOCITY OF THE ROBOT IN A SIMULATION WITH GAIN SCHEDULING

In this simulation, the robot was initially traveling at a forward velocity of V = 0.16 m/s and at t = 10s, slowly increasing its velocity. At t = 40s, the vehicle reached approximately 0.49m/s which it maintained for another 10 seconds. It then began to increase its velocity and reached a final velocity of 0.75 m/s at t = 80s, which it maintained until the end of the simulation. Disturbances were applied to the robot during the time it was transitioning between the steady state velocities. At t = 15s, a moment disturbance in the yaw direction (approximately 0.25Nm) was applied for 5 seconds and a pitch moment disturbance was applied at t = 65s lasting for 5 seconds. As the figure indicates, the robot was able to recover from the disturbances very well and was able to maintain its velocity.

Experimental Results

The stability augmentation system was implemented on the actual robot in a swimming pool and in open sea trials (shown in FIG. 10). In each of the trials, the gains used and the data from the inertial measurement unit (IMU) were logged. Unfortunately, the yaw angle measured using the magnetic compass was not reliable due to the noise created by the other electronics within the robot and thus the gain on the yaw angle was set to $K_\psi = 0$ for all trials.

In the pool trials, AQUA swam back and forth and occasionally a swimmer applied a moment disturbance on the robot by pushing on the robot. After the disturbance was applied, the robot was allowed to respond without any further interference.

This was done multiple times with various gains acting on the angles and angular rates of the robot.

FIG. 10 AQUA IN SEA TRIALS

An example of a set of trials with gains K_q = 1N·s/rad, K_ϕ = 4N/rad, and K_θ = 8N/rad can be seen in FIG. 11. The solid line represents the response without a SAS and it shows that after a disturbance was introduced at t = 8s, the robot's angles were not able to return to zero. The dashed line is the response with SAS and it illustrates that the robot was able to swim forward with only small oscillations and that it was able to recover quickly from external perturbations at t = 4s and 8s.

A qualitative evaluation was done by an expert pilot making the vehicle swim back and forth in a grid pattern and record video of a coral reef below. This

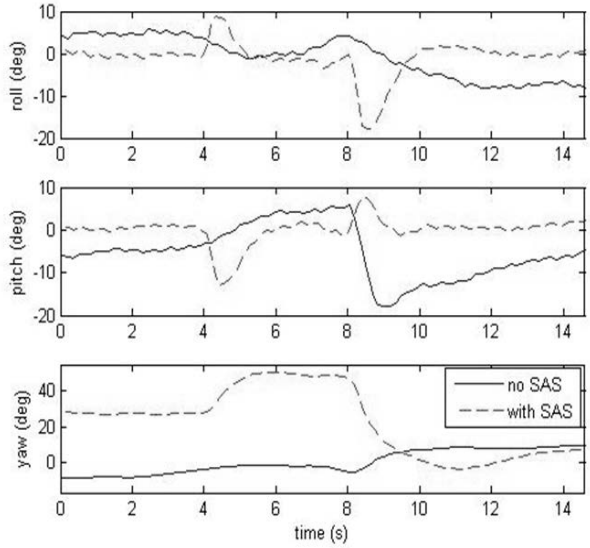

FIG. 11 EXPERIMENTAL RESULTS WITH GAINS ACTING ON THE ROLL AND PITCH ANGLES

driver usually sat in a small boat and drove the robot from the surface of the water. Therefore, he often had to contend with a rocking boat and poor visibility while trying to keep the robot steady as it swam over the reef. He found that the SAS was a considerable improvement. By having the robot swimming steadily on its own, the driver was able to concentrate more on obtaining the video images he wanted.

Fault Tolerance

AQUA is meant to be used in conditions that may not be safe for human divers. Long missions and unpredictable environments may lead to failures in the robot, including loss of mobility or outright loss of one of its flippers. These failures can be caused by internal malfunctions, such as a motor breakdown, or by external interferences, such as collisions with other objects. Ideally, the SAS designed in the previous section would be robust enough to compensate for the failures, but this turned out not to be the case with both types of failures. In this section the SAS is reconfigured and the approach used assumed that the existence of a failure could be accurately detected. However, since a single modified SAS is able to handle all the failures considered, it is not necessary to know which leg has failed.

Missing Flipper

It was found through simulations and experiments that the SAS designed in the previous section was able to compensate for the case of missing any one of the flippers. FIG. 12 displays the results from one such set of experiments. The solid line illustrates part of one trial where the SAS was not used and it is evident that

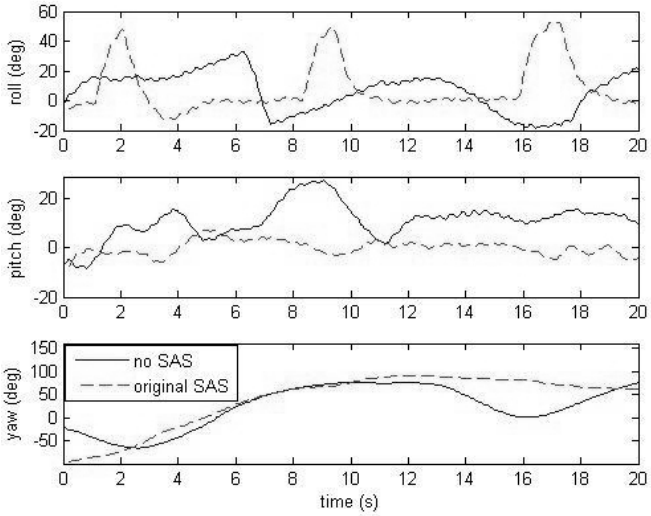

FIG. 12 EXPERIMENTAL RESULTS WITH THE BACK LEFT LEG MISSING

the roll and pitch angles could not be maintained at

zero degrees. From the beginning of the data shown, the roll angle was deviating from zero and at t = 6s the pilot attempted to correct the roll angle. Afterwards, it can be seen that the roll angle began to rise again. Similarly, the pitch angle was not constant and by t = 9s angle was fairly large. The pilot attempted to correct the pitch angle, after which the pitch angle rose again but then stayed somewhat constant around 10 degrees with no further commands from the pilot.

In contrast, the dashed line in FIG. 12 represents a trial where the SAS was used and it shows a great improvement in the behavior of the robot. Roll disturbances were applied to the robot at t = 1s, 8s and 16s and each time, the roll angle returned to zero within 1 second after the disturbance was removed, with a small overshoot after the first disturbance. The time between the disturbances also shows that the SAS is able to maintain the roll angle at ±2 degrees and the pitch angle at ±5 degrees.

Stuck Flipper

The next fault investigated was the case of a flipper being stuck at a fixed angle. The flipper angle, α, is considered to be zero when the flipper points upward; -π/2 when it points directly to the rear; and -π when it points downwards. The goal was to design a SAS that would compensate for the failure without knowing what angle the leg was stuck at or which leg was stuck.

Simulation of the robot was done for various legs stuck at various angles between 0 and -π, since AQUA's flippers stay within this range in normal operations. In these simulations, the original SAS was implemented and there were no external disturbances. FIG. 13 shows the effect of the back left leg stuck at various angles. Similar responses were found with the other five legs stuck at various angles and will be discussed later in this section.

From the figure, it appears that the SAS is unable to compensate for a stuck leg. The robot did not become unstable but settled to a different steady state with a large orientation error. In most cases, the steady state error is larger than 10 degrees, which is not an acceptable error. In addition to the loss of the force in the x-direction to keep a steady forward velocity, there is also an added drag force acting on the robot. The combination of these two effects is large enough that the baseline SAS cannot compensate for it. Since all the angles are strongly coupled to one another, when one angle deviated from zero, the others did as well.

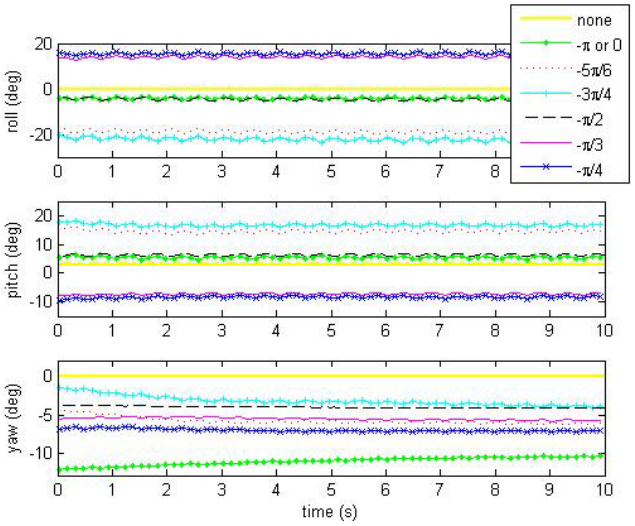

FIG. 13 RESPONSE OF AQUA TO THE BACK LEFT LEG STUCK AT VARIOUS ANGLES USING THE ORIGINAL SAS

FIG. 13 also shows that the roll and pitch were most affected when a flipper was stuck at -π/4 or -3π/4. And the yaw angle was most affected when the flipper was stuck at 0 or -π. Since the greatest effects were observed at these angles, the design of a new SAS was based on them. It was assumed that if the new SAS could compensate for these extreme cases, it could also compensate for other intermediate angles.

To begin, the drag forces acting on the robot in the extreme cases (flipper angles -π and -3π/4) were examined. When the flipper is at -π, the drag force acts only in the x-direction and can be estimated as:

$$F_d = \frac{1}{2}\rho v^2 C_d A \qquad (11)$$

where ρ = 1000kg/m^3 is the density of water, v = 0.49m/s is the forward velocity of AQUA, C_d = 1.2 is the drag coefficient (taken from Geordiades, 2005), and A = 0.014 m^2 is the area of a flipper.

Based on this simplified analysis, the drag force created by the flipper when it is perpendicular to the forward velocity of the robot is approximately 2.1N. FIG. 14 illustrates the case where the back left flipper is stuck and shows the drag force created by the flipper. Since the flippers are approximately 0.185m from the x-axis of the robot frame, a moment of 0.389N·m is produced and it causes the robot to yaw.

The reconfigured SAS aims to have the remaining working five flippers producing an equal and opposite moment to compensate for the fault. Assuming the other five flippers are also 0.185m away from the x-axis, then each of the five flippers must change its thrust by 0.42N in the x-direction, as shown in FIG. 14, to counteract the drag moment.

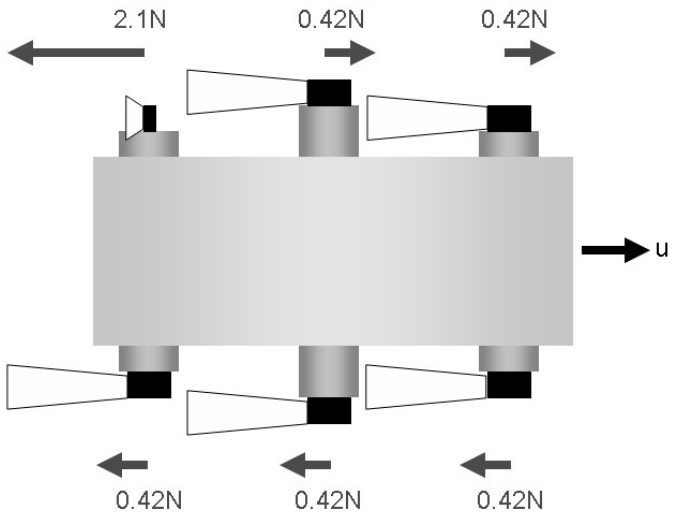

FIG. 14 FORCES CONTRIBUTED BY EACH FLIPPER

To calculate the gain needed to do this, the five functional flippers needed to provide 0.42N when the yaw angle was at some acceptable steady state error. That error was based on the results from the previous section, where the yaw angle remained between 2-4 degrees when the SAS was used. Thus with an error of 2 degrees or π/90 rad, the extra gain that needed to be added to the existing K_ψ was calculated as:

$$\Delta K_\psi = \frac{F_d}{\pi/90} = 12 \text{ N/rad} \qquad (12)$$

To determine the extra gain needed for ΔK_ϕ and ΔK_θ, a similar process was followed. The gains on the angular rates were also investigated but it was found that they did not affect the results significantly. Thus, ultimately only the gains on the roll, pitch and yaw angles were changed.

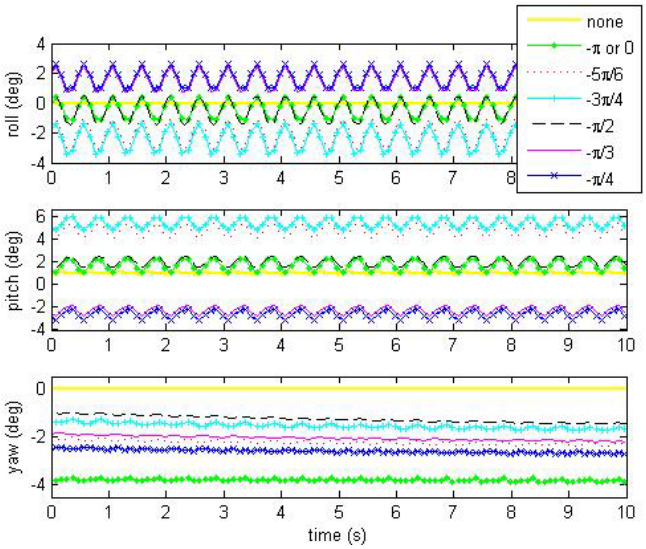

FIG. 15 RESPONSE OF AQUA TO THE BACK LEFT LEG STUCK AT VARIOUS ANGLES USING THE NEW SAS

Simulations were run again with the back leg stuck at various angles. FIG. 15 shows the results using the same angles as in FIG. 13, and there is a definite improvement in the angular response. The angles were reduced to ±4deg which was considered acceptable. The modified SAS was used with the other flippers stuck at -3π/4rad with similar successful results. Thus, these results demonstrate that the modified SAS does not need to take into account which leg was stuck, nor the angle at which the leg was stuck.

Experimental Results

The modified SAS was implemented on the physical robot in pool trials. As before, the gain on the yaw angle was K_ψ = 0 for all the trials, since the measured yaw angle was not reliable. AQUA swam back and forth along the length of the pool and disturbances were applied by the pilot using the game pad, but only when a steady state could be reached by the robot.

A set of trials was performed with the back left leg fixed at -3π/4rad, shown in FIG. 16. The solid line represents the response of the robot without a SAS. In this trial, no disturbances were applied since the robot could not reach a steady state. At t = 1.5s, 8s and 15s corrective commands were given by the pilot because the roll angle was becoming too large. After each correction, the robot drifted away from zero immediately and continued to drift. The behavior of the pitch angle was also erratic and varied between -20 and 45 degrees.

The response of the robot is greatly improved with the use of the original SAS, which is represented by the dashed line. Roll disturbances were applied at t = 1.5s, 5.5s, and 10.5s and the robot was able to recover from the disturbance within a second after the disturbance was removed. AQUA was also able to recover from a pitch disturbance applied at t = 15.5s. Smaller disturbances appear in the pitch angle whenever a roll disturbance was applied, which indicates that the pitch angle is strongly affected by changes to the roll angle and vice versa.

Finally, the modified SAS was implemented on AQUA but the K_θ used was lower than the value calculated in the preceding section. Here, roll disturbances were applied at t = 0.5s, 5.5s, 11s and 17s with varying durations. The figure clearly shows that the robot was able to recover within a second after each disturbance was applied. The roll response of the robot with the modified SAS is similar to the response with the

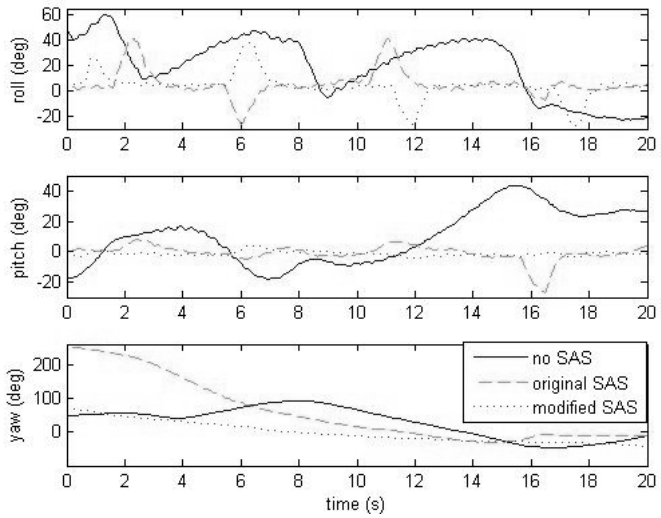

FIG. 16 EXPERIMENTAL RESPONSE OF THE ROBOT WITH THE BACK LEFT LEG FIXED AT -3π/4 rad

original SAS. However, a difference can be seen in the pitch response. With the original SAS, the pitch angle varied by about ±8 degrees but with the modified SAS, the response was smoother with the angle varying by approximately ±5 degrees.

Conclusions

In this work, a linear model of a hexapod underwater robot was derived using numerical differentiation. The state space matrices were developed for three different steady state velocities and were used to design a stability augmentation system. The gain matrices were refined using the non-linear model. Gain scheduling was then implemented to allow the SAS to operate throughout the vehicle's speed range. The resulting system demonstrated good recovery from external disturbances in simulation, in pool trials and in open sea experiments.

Two different failures were then investigated: the loss of a flipper and a stuck flipper due to a motor failure. The original SAS was found to be robust enough to compensate for the case of a lost flipper, but not for the case of the stuck flipper. A modified SAS was designed and shown to work both in simulation and in experiments. It was not necessary to know which leg had failed, nor the angle at which the failed flipper was stuck.

ACKNOWLEDGMENT

The authors gratefully acknowledge the financial support of the Natural Sciences and Engineering Research Council for this work.

REFERENCES

Do, Khac D., Zhong-Ping Jiang, Jie Pan, and Henk Nijmeijer. "A Global output-feedback Controller for Stabilization and Tracking of Underactuated Odin: A Spherical Underwater Vehicle." Automatica 40 (2004): 117–124.

Fossen, Thor I., and Jens Balchen. "Modelling and Non-linear Self-tuning Robust Trajectory Control of an Autonomous Underwater Vehicle." Modelling, Identification and Control 9 (1988): 165–177.

Fossen, Thor I. Guidance and Control of Ocean Vehicles, UK: John Wiley & Sons Ltd, 1994.

Fullmer, Rees. R., Robert Gunderson, and Gordon Olsen. "The Preliminary Design of a Stability Augmentation System for a Remotely Piloted Vehicle in Hover." IEEE Conference on Control Applications, Vol. 2: 825–826, 1992.

Georgiades, Christina et al. "AQUA: An Aquatic Walking Robot." IEEE/RSJ International Conference on Robotics and Systems, Vol. 4: 3525–3531, 2004.

Georgiades, Christina. "Simulation and Control of an Underwater Hexapod Robot." Master's thesis, McGill University, 2005.

Georgiades, Christina, Meyer Nahon, and Martin Buehler. "Simulation of an Underwater Hexapod Robot." Ocean Engineering 36 (2009): 39-47.

Healey, Anthony, Stephen Rock, Steven Cody, D. Miles, and James Brown. "Toward an Improved Understanding of Thruster Dynamics for Underwater Vehicles." IEEE Journal of Oceanic Engineering 20 (1995): 354–361.

Hsu, Stephen, Chris Mailey, Ethan Eade, and Jason Janét. "Autonomous Control of a Horizontally Configured Undulatory Flap Propelled Vehicle." Proceedings of the IEEE International Conference on Robotics & Automation, 2194–2199, 2003.

Kahn, Aaron D. "Attitude Command Attitude Hold and Stability Augmentation Systems for a Small-scale Helicopter UAV." Proceedings of Digital Avionics Systems Conference, Vol. 2: 8.A.4/1–8.A.4/10, 2003.

Lea, Roy, Robert Allen, and Stephanie Merry. "A Comparative Study of Control Techniques for an Underwater Flight Vehicle." International Journal of Systems Science 30 (1999): 947–964.

Leonard, Naomi E. "Control Synthesis and Adaptation for Underactuated Autonomous Underwater Vehicle." IEEE Journal of Oceanic Engineering 20 (1995): 211–220.

Licht, Stephen, Franz Hover, and Michael Triantafyllou. "Maneuvering 'Finnegan the Roboturtle' Through Force Vectoring with Oscillating Foils." Proceedings of the International Symposium on Unmanned Untethered Submersible Technology, 362–369, 2007.

Nakamura, Yoshihiko, and Shrikant Savant. "Nonlinear Tracking Control of Autonomous Underwater Vehicles." Proceedings of IEEE International Conference on Robotics and Automation, A4–A9, 1992.

Ni, Lingli, and Chris Fuller. "Control Reconfiguration Based on Hierarchical Fault Detection and Identification for Unmanned Underwater Vehicles." Journal of Vibration and Control 9 (2003): 735–748.

Oliva, Alvaro P. "An Altitude Hold Autopilot Design to Work with an Observer-based Stability Augmentation Control Law." Proceedings of IEEE Conference on Control Applications, Vol. 1: 353–358, 1994.

Orrick, Alec, Make McDermott, David Barnett, Eric Nelson, and Glen Williams. "Failure Detection in an Autonomous Underwater Vehicle." Proceedings of the IEEE Symposium on Autonomous Underwater Vehicle Technology, 377–382, 1994.

Rae, Graeme, and Stan Dunn. "On-line Damage Detection for Autonomous Underwater Vehicles." Proceedings of the IEEE Symposium on Autonomous Underwater Vehicle Technology, 383–392, 1994.

Saranli, Uluc, Martin Buehler, and Daniel Koditschek. "Rhex: A Simple and Highly Mobile Hexapod Robot." The International Journal of Robotics Research 20 (2001): 616–631.

Smallwood, David A., and Louis Whitcomb. "Model-based Dynamic Positioning of Underwater Robotic Vehicles: Theory and Experiment." IEEE Journal of Oceanic Engineering 29 (2004): 169–186.

Yang, Keith, Junku Yuh, and Song Choi. "Fault-tolerant System Design of an Autonomous Underwater Vehicle Odin: An Experimental Study." International Journal of Systems Science 30 (1999): 1011–1019.

Yuh, Junku. "Design and Control of Autonomous Underwater Robots: A Survey." Autonomous Robots 8 (2000): 7–24.

Olivia Chiu obtained a B.A.Sc. in Engineering Physics from Queens University in Kingston, and a Masters degree in Mechanical Engineering from McGill University in Montreal.

She was a masters student in the Department of Mechanical Engineering at McGill University from 2006 to 2008. Since then, she has been a staff software engineer at National Instruments in Texas. Her research interests focus on software and hardware development for autonomous systems.

Meyer Nahon obtained a B.A.Sc. in Mechanical Engineering from Queen's University in Kingston, an M.A.Sc. in Aerospace Engineering from the University of Toronto, and a Ph.D. in Mechanical Engineering from McGill University in Montreal.

He was a faculty member in the Department of Mechanical Engineering at the University of Victoria from 1991 to 2001. Since then he has been at McGill University, presently as a Professor of Mechanical Engineering. His research deals with various aspects of dynamics and control of aerial and marine vehicles and systems, with a particular emphasis on tethered systems.

Prof. Nahon is a Senior Member of the IEEE, and an Associate Fellow of the AIAA and Fellow of the Canadian Aeronautics and Space Institute (CASI). He has received awards from the AIAA and CASI for his work on flight simulator motion systems and on space-based robotics.

Nicolas Plamondon obtained a B.A.Sc. in Mechanical Engineering from the University of Toronto, and a Ph.D. in Mechanical Engineering from McGill University in Montreal.

He was a doctoral student in the Department of Mechanical Engineering at McGill University from 2005 to 2010, working on controller development for the AQUA vehicle. Since 2010, he has been an analyst in control systems at Pratt & Whitney Canada. His research interests relate to control design and autonomous systems.